

논문 2010-47SD-9-1

적응적 파이프라인을 적용한 저전력 H.264 복호기 설계

(Design of Low Power H.264 Decoder Using Adaptive Pipeline)

이 찬 호*

(Chanho Lee)

요 약

H.264 영상 압축 표준은 높은 압축률과 화질로 널리 이용되고 있다. H.264 복호기는 일반적으로 매크로블록 또는 4x4 하위블록 단위로 파이프라인을 적용하여 동작한다. 이러한 파이프라인 한 단의 주기는 보통 최악의 상황에서도 동작을 보장하도록 결정되어 높은 전송 대역폭과 고성능 연산기를 요구하고 연산기가 일을 하지 않고 쉬는 사이클이 많아지는 결과를 초래한다. 본 논문에서는 이러한 연산기의 쉬는 사이클을 줄이고 데이터 전송 대역폭과 연산기 성능 요구 조건을 완화시킬 수 있는 적응적 파이프라인 구조를 채택한 효율적인 영상 복호기 구조를 제안한다. 제안한 구조에서는 파라미터와 계수는 핸드셰이킹 방식으로 전용 신호선을 통해 전달되고 복호된 영상 데이터는 AMBA AHB 네트워크를 통해 메모리에 저장하거나 읽어 온다. 각 블록의 복호 처리 시간은 영상의 특성에 따라 가변적으로 변하고 각 연산기는 데이터가 준비되면 언제든지 동작을 할 수 있다. 제안한 구조에 따라 H.264 복호기를 설계하였고 FPGA를 이용하여 동작을 검증하였다.

Abstract

H.264 video coding standard is widely used due to the high compression rate and quality. H.264 decoders usually have pipeline architecture by a macroblock or a 4 x 4 sub-block. The period of the pipeline is usually fixed to guarantee the operation in the worst case which results in many idle cycles and the requirement of high data bandwidth and high performance processing units. We propose adaptive pipeline architecture for H.264 decoders for efficient decoding and lower the requirement of the bandwidth for the memory bus. Parameters and coefficients are delivered using hand-shaking communication through dedicated interconnections and frame pixel data are transferred using AMBA AHB network. The processing time of each block is variable depending on the characteristics of images, and the processing units start to work whenever they are ready. An H.264 decoder is designed and implemented using the proposed architecture to verify the operation using an FPGA.

Keywords : H.264, 복호기, 적응적 파이프라인, 핸드셰이킹, AHB, FPGA

I. 서 론

H.264 영상 압축 표준은 ISO/IEC와 ITU-T가 함께 JVT(Joint Video Team)를 구성하여 개발하였으며 시장이 고화질의 동영상 요구함에 따라 가장 널리 사용되는 압축 표준으로 자리 매김하고 있다. H.264는 MPEG-4 part 10 AVC (Advanced Video Coding)로도

불리는데 MPEG-4 Visual의 ASP (Advanced Simple Profile)보다 25-40% 정도 압축률이 좋다.^[1~2] 이 압축 표준은 가변 블록 크기와 1/4 픽셀 보간 예측, 인트라 예측 (Intra-Prediction: IP), CAVLC (Context-based Adaptive Variable Length Coding)과 CABAC (Context-based Adaptive Binary Arithmetic Coding)과 같은 새로운 특징을 포함한다. 이러한 특징은 압축률을 높이고 고화질의 영상을 제공하지만 복잡한 연산으로 높은 연산기 성능을 요구한다. H.264 복호기의 연산기 복잡도는 일반적으로 MPEG-4에 비해 2배 정도 복잡하다.

H.264 복호기는 데이터 처리 특성상 일반적으로 매크로블록 (Macroblock)이나 4x4 하위블록 (sub-block)

* 정회원, 숭실대학교 정보통신전자공학부
(School of Electronic Engr., Soongsil University)
* 본 논문은 지식경제부 출연금으로 ETRI 시스템반도체진흥센터에서 시행한 시스템반도체 융복합형 설계인재양성사업의 연구결과이며 IDEC의 SW지원을 받았습니다.
접수일자: 2010년4월20일, 수정완료일: 2010년8월23일

단위로 파이프라인 구조를 갖는다. 이때 이러한 블록 단위로 부호화 방식이 달라질 수 있으므로 일반적으로 영상의 특성에 따라 블록마다 복호 시간이 달라진다. 파이프라인의 한 주기를 고정시킬 경우 가장 연산 시간이 긴 조건을 기준으로 결정되어야 하므로 높은 동작 주파수의 고성능 하드웨어가 요구된다. 그러나 그러한 최악의 상황은 자주 발생하지 않으므로 한 주기에 연산기가 동작을 끝내고 쉬는 사이클이 많아지게 된다. 이러한 쉬는 사이클을 모두 제거한다면 더 낮은 성능의 연산기로도 동일한 복호 과정을 진행할 수 있다.

H.264 복호 소프트웨어를 이용하여 복호과정을 실시간 분석해보면 움직임 보상기 (Motion Compensation: MC)가 전체 복호시간의 55% 정도를 차지하는 것을 볼 수 있다.^[3-4] 따라서 움직임 보상기가 일반적으로 복호기 파이프라인의 병목이 된다. 움직임 보상기의 성능은 두 가지 요소에 의해 결정된다. 하나는 보간 연산기의 연산 능력이고 다른 하나는 참조 메모리에서 픽셀 데이터를 읽어 오는 시간이다. H.264 복호기에는 참조 메모리에 접근하는 3개의 장치가 있는데 움직임 보상기와 디블록킹 필터 (Deblocking Filter: DF), 그리고 영상 데이터 출력 장치이다. 영상 데이터 출력 장치는 복호와 직접적인 관련은 없지만 영상 데이터를 화면이나 호스트 시스템으로 보내기 위해 필요하다. 일반적으로 참조 메모리는 하나의 DRAM 칩으로 구성되어 한 번에 오직 하나의 장치만 메모리에 접근할 수 있으므로 이들 세 장치는 참조 메모리 접근을 위해 서로 경쟁을 하게 된다. 이 경쟁에서 적절하게 메모리 접근을 조절하지 않으면 복호기의 성능이 저하된다.

본 논문에서는 H.264 복호기를 위한 효율적인 하드웨어 구조를 제안한다. 제안하는 구조는 복호기 파이프라인의 주기가 영상 부호화 방식에 따라 달라져 연산기가 쉬는 사이클을 최소화하여 보다 저성능의 연산기로 복호가 가능하게 하는 적응적(adaptive) 파이프라인 방식을 가지고 있다. 파이프라인을 통한 데이터 전달은 핸드셰이킹 방식으로 이루어져 별도의 제어가 필요 없고 연산기가 준비가 되는 즉시 데이터 처리가 이루어지므로 효율적인 복호가 가능하다. 이러한 구조는 데이터가 없을 때 불필요한 동작을 막아 전력 소모를 감소시키는 효과도 존재한다. 또한 참조 메모리 접근에서의 서의 성능 저하를 막기 위해 중재 DMAC (Direct Memory Access Controller with Arbiter: DMACA)가 존재하여 성능을 최대화하는 방향으로 메모리 접근 제

어를 실행한다. 이러한 메모리 접근 제어만으로 전체 파이프라인을 제어할 수 있어 복잡한 제어기 없이 복호기를 구성할 수 있고 연산기의 확장 및 개선, 교체가 용이하다. 제안한 구조에 따라 H.264 복호기를 설계하여 FPGA를 이용하여 구현하고 그 동작을 검증하였다.

II. 제안하는 H.264 복호기 구조

1. NAL 입력과 복호 과정

부호화기는 영상 데이터를 압축하여 여러 개의 NAL (Network Abstraction Layer) 유닛 조합으로 구성한다. 따라서 복호기는 NAL 비트 스트림을 입력으로 받는다. H.264 복호기는 일반적으로 가변 길이 복호기 (Variable Length Decoder: VLD), 역변환 및 양자화기 (Inverse Transform and Quantization: ITQ), 인트라 예측기 (IP), 움직임 보상기 (MC), 그리고 디블록킹 필터 (DF)로 구성된다. VLD는 비트 스트림을 받아 양자화된 계수와 파라미터를 생성한다. 파라미터는 모든 연산 유닛으로 보내지고, 양자화된 계수는 ITQ로 보내진다. ITQ는 받은 계수들을 역양자화와 역변환 과정을 거쳐 오차 값을 생성한다. MC와 IP는 VLD에서 보내온 파라미터를 이용하여 예측 블록을 생성한다. MC는 참조 프레임 메모리의 픽셀 데이터 F'_{n-1} 를 이용하고 IP는 동일 프레임내의 이전 블록 데이터를 이용한다. MC/IP에서 생성된 예측 블록과 ITQ에서 생성된 오차 값을 더하면 픽셀 데이터가 만들어진다. 이렇게 만들어진 데이터는 DF로 보내져 필터링을 통해 복호화된 블록 F'_n 을 생성한다.

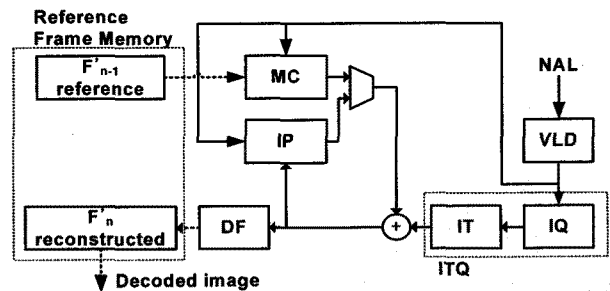


그림 1. H.264 복호기의 복호과정

Fig. 1. Decoding procedure of H.264 Decoder.

2. 적응적 파이프라인 구조

그림 1에서 데이터 전달 경로중 실선으로 표시된 부분은 NAL 데이터로부터 시작하여 연산기를 통하는 경

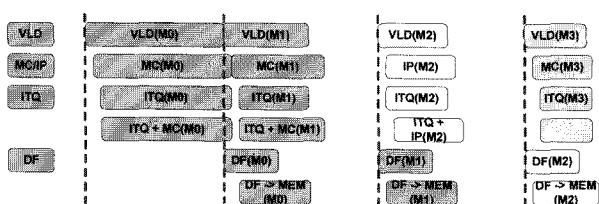


그림 2. 고정 주기의 파이프라인 동작 순서도
Fig. 2. Processing sequence of decoder pipeline with a fixed period.

로이고 점선은 참조 프레임 메모리를 접근하는 경로이다. 일반적으로 실선으로 표시된 경로를 중심으로 파이프라인이 구성된다. 각 연산기는 매크로블록 또는 4x4 하위 블록을 단위로 연산을 수행하는데 매크로블록에 따라 부호화 방식이 다르므로 연산 시간에 차이가 있다. 이 때 각 파이프라인의 주기를 고정할 경우 가장 복잡한 복호 방식에 맞추어야 파이프라인이 항상 동작하므로 주기가 길어진다.^[5] 또한 파이프라인을 제어하는 복호 제어가 필요하다. 그림 2에 고정 주기의 파이프라인 동작 순서가 나타나 있다. 매크로블록 M0의 연산 시간이 가장 길다고 할 때 주기는 M0 연산이 맞추어 결정되고 이후 연산 시간이 짧은 M1 이후의 매크로블록도 동일한 주기를 가지고 연산이 진행된다. 따라서 M1 이후에는 연산기가 쉬는 사이클이 포함된다.

적응적 파이프라인은 이러한 쉬는 사이클을 제거하여 복호 방식에 따라 파이프라인 주기를 가변적으로 변경할 수 있는 방식이다. 그림 3에 나타난 바와 같이 쉬는 사이클을 제거하여 동일한 시간에 더 많은 블록에 대한 복호를 진행할 수 있다. 이는 영상의 특성에 따라 부호화 방식이 달라져 매크로블록별로 복호시간도 달라지는 것을 이용한 것이다. 따라서 매크로블록마다 복호 시간을 감시하여 다음 매크로블록 처리가 계속해서 이어지도록 정교한 제어를 해야 한다.

그림 4는 적응적 파이프라인을 적용한 경우 하나의

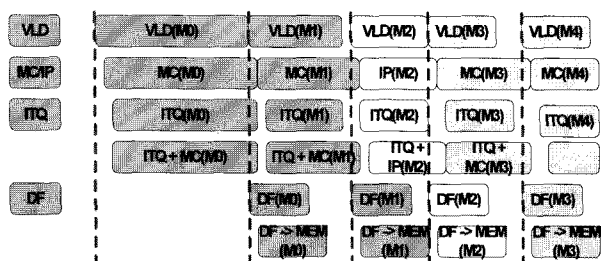


그림 3. 가변 주기의 적응적 파이프라인 동작 순서도
Fig. 3. Processing sequence of adaptive pipeline with variable period.

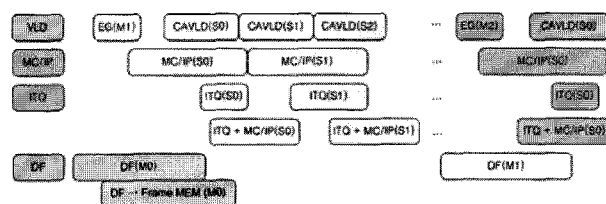


그림 4. 적응적 가변 파이프라인을 적용한 복호 흐름도
Fig. 4. Adaptive Variable pipeline in the proposed decoder architecture.

매크로블록을 4x4 하위 블록 단위로 처리하는 과정을 나타내고 있다. VLD에서 Exp-Golomb (EG) 복호기가 파라미터를 추출하여 MC/IP로 보내면 예측 연산이 시작되고 4x4 하위 블록 단위로 CAVLD에서 계수를 복호하여 ITQ로 보내면 역양자화 및 역변환 과정이 시작된다. 예측 연산과 역변환 결과는 더해진 뒤 하위 블록 단위로 DF의 버퍼에 저장된다. DF는 하나의 매크로블록에 해당하는 데이터가 모이면 필터 연산을 진행한다. 일반적으로 DF 연산은 하나의 매크로블록만큼 밀려서 진행된다. DF 역시 하위 블록 단위로 연산을 진행할 수 있지만 제어가 복잡해지는 반면 첫 번 매크로블록의 출력 잠복기가 조금 빨라지는 것 외에 성능면에서 큰 이득은 없다.

적응적 파이프라인이 복호시간을 전반적으로 줄여주는 장점이 있는 반면에 전체 흐름에 대한 정교한 제어가 필요하여 제어가 복잡해질 수 있는 단점이 있어 이를 극복할 수 있는 복호기 구조가 필요하다. 또 다른 문제점은 주기가 일정하지 않아 참조 메모리 접근에 필요한 시간이 균등하게 할당이 안 되므로 메모리 접근에 대한 스케줄링이 효과적으로 이루어져야 한다. 이 또한 제어를 복잡하게 하는 요인이므로 이를 해결할 수 있는 복호기 구조가 필요하다.

3. 복호기 구조

본 논문에서 제안하는 복호기 구조는 적응적 파이프라인을 적용하여 불필요한 유휴 사이클을 줄여 효율성을 높이면서 제어가 복잡해지는 문제점을 완화한 특징을 가지고 있다. 위에서 설명한 바와 같이 그림 1에서 실선 경로는 파라미터 또는 계수가 순차적으로 전달되는 파이프라인 구조를 형성한다. 제안하는 구조에서는 각 연산기에서 데이터가 전달되는 동작을 복호기의 중앙 제어기에서 하지 않고 그림 5와 같이 연산기 사이를 핸드셰이킹 방식으로 연결하여 각 연산기는 제어기의 지시 없이 연산 준비가 되면 데이터를 받아 연산을

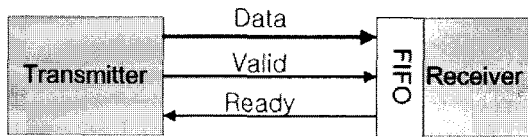


그림 5. 연산기 사이의 핸드셰이킹 통신

Fig. 5. Hand-shaking communication between processing units.

시작하고 연산이 완료되면 다음 연산기로 전달한다. 따라서 입력된 NAL 데이터는 파이프라인 구조에 의해 DF까지 순차적으로 이동하여 복호된 영상 이미지가 만들어진다. 이 과정은 중앙 제어기의 제어를 받지 않는다. 복호 흐름의 제어는 NAL 데이터의 공급을 제어함으로써 가능하다. 각 연산기는 입력 데이터가 준비되고 이전 데이터에 대한 연산이 끝난 뒤에는 즉시 새로운 데이터를 받아 연산을 진행하고 연산 완료후 다음 연산기로 보낸다. 이때 다음 단 연산기의 FIFO가 비어 있지 않으면 기다린다. 이러한 적응적 가변 파이프라인은 부호화 방식에 따라 매크로블록마다 달라지는 복호 방식을 적용하여 연산이 진행되고, 연산이 완료되면 즉시 다음 연산을 진행하므로 항상 하나 이상의 연산기가 동작하는 상태를 유지할 수 있다. 또한 데이터가 공급되지 않는 연산기는 동작하지 않으므로 불필요한 동작을 하지 않으므로 전력 소모를 줄일 수 있다.

그림 1에서 점선으로 표시된 경로는 참조 메모리에 대한 접근을 나타낸다. 서론에서 언급한 바와 같이 참조 메모리는 MC, DF와 영상 출력 장치 등 세 개의 유닛이 접근하는데 일반적으로 단일 칩의 DRAM으로 구성되므로 한 번에 하나의 읽기 또는 쓰기 동작만 가능하다. 일반적으로 MC가 복호기 파이프라인의 병목이므로 MC가 참조 메모리에 접근할 때 우선권이 주어지야 복호기의 성능을 향상시키는데 도움이 된다. 그러나 MC에만 계속 우선권을 줄 경우 DF가 필터링이 끝난 복호 영상을 기록하지 못하면 파이프라인이 멈추게 되므로 MC와 DF는 적절하게 우선권을 조절하여 MC의 동작을 방해하지 않으면서 파이프라인이 계속 동작하도록 해야 한다. 영상 출력 장치는 상대적으로 우선권이 낮아도 큰 문제가 되지 않지만 계속 참조 메모리 접근을 못하면 두 가지 문제가 발생한다. 첫째는 복호된 데이터가 정해진 속도(frame rate)로 출력 장치에 전달되어야 원본과 같은 영상을 보여주는데 이를 맞추지 못하면 화면이 끊어지는 느낌이 들거나 느린 동작을 보여준다. 둘째는 참조 메모리에는 다수의 참조 픽셀 프레임

데이터가 저장되는데 가장 오래된 프레임 데이터부터 새로운 데이터로 덮어 쓴다. 그런데 영상 출력 장치가 계속 참조 메모리에 접근을 못하여 가장 오래된 프레임 데이터가 출력되지 않은 상태에서 덮어 쓰이면 해당 프레임은 출력되지 않고 사라진다. 따라서 두 가지 문제가 발생하지 않도록 우선권을 조절할 필요가 있다.

제안한 구조에서는 DMACA가 영상출력 장치를 포함하면서 MC와 DF의 요청을 받아 참조 메모리를 접근하여 데이터를 쓰거나 읽는 동작을 하도록 하였다. 이때 DMACA는 참조 메모리 상태와 영상 출력 속도를 감시하면서 어느 요청을 먼저 처리할지를 결정하는 제어기 역할을 하면서 복호기가 최적의 성능을 내도록 한다. 그러나 복호가 진행되는 파이프라인의 동작에는 전혀 관여하지 않는다.

III. H.264 복호기 설계 및 검증

제안한 적응적 가변 파이프라인과 DMACA를 적용하여 복호기를 설계하였다. 그림 6에 그 구조가 나타나 있다. NAL 데이터는 별도의 인터페이스를 통해서 공급되고 VLD, ITQ-MC/IP와 DF를 거치는 적응적 파이프라인을 핸드셰이킹 방식으로 전용 신호선을 통해 파라미터와 계수가 전달된다. 내부 SRAM은 IP와 DF가 이웃 매크로블록 데이터를 임시로 저장하기 위한 것이다. MC는 보간 연산을 위해 참조 메모리에서 읽기 접근이 필요하고 DF는 현재 프레임 데이터를 참조 메모리에 저장하기 위해 쓰기 접근이 필요하다. 두 작업은 MC와 DF가 DMACA에 인터럽트 신호를 보내 이루어진다. 참조 메모리 접근을 위해 AHB 네트워크를 통해 마스터인 DMACA와 슬레이브인 SDRAM 제어기와 MC, 그리고 DF가 연결된다. DMACA 내부에는 출력 장치로 데이터를 내보기 위한 data_out 장치가 포함되고 DMACA를 통해 참조 메모리의 픽셀 데이터를 읽어 온다.

설계한 복호기에서 VLD는 Exp-Golomb 복호기와 CAVLD로 구성되는데 Exp-Golomb 복호기는 순수한 연산기로 구성되어 있으며, CAVLD는 테이블과 연산기를 혼합하여 최적화된 구조를 갖고 HD 영상의 실시간 복호가 가능하다.^[6]

ITQ-MC/IP는 인터페이스는 공유하고 있으나 내부적으로는 별도의 연산기로 구성되어 있으며 VLD에서 파라미터가 전달되면 슬라이스와 매크로블록 형태에 따

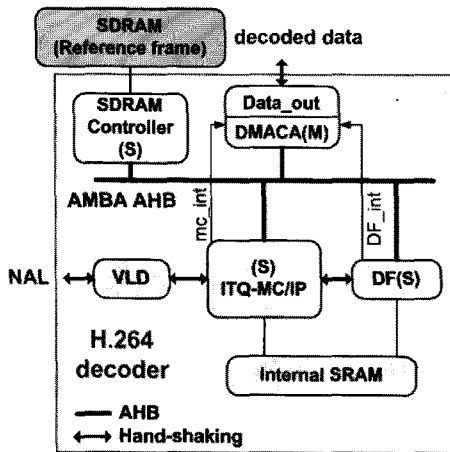


그림 6. 제안한 구조에 따른 H.264 복호기 구조
 Fig. 6. Block diagram of H.264 decoder according to the proposed architecture.

라 MC 또는 IP가 동작을 한다. 이때 MC와 IP는 파라미터만 적절하게 전달되면 동시에 연산이 가능하다. IP는 I 슬라이스 또는 매크로블록을 처리하고 스케일러블한 구조를 가지고 있어 내부 연산기의 확장을 통해 필요한 연산 능력을 얻기 쉬운 구조이다.^[7] MC는 P 슬라이스 또는 매크로블록을 처리하며 2 차원 회전 레지스터 파일과 움직임 벡터 예측기, 그리고 저복잡도 고성능의 보간 연산기를 이용한 효율적인 구조를 갖고 있다.^[8] ITQ는 계수를 받아 역양자화 및 역변환 과정을 거쳐 오차 픽셀 데이터를 생성시키는데 연산 순서를 조절하고 데이터 선처리를 통해 내부 구조를 단순화한 구조를 적용하였다.^[9] ITQ와 MC 또는 IP의 결과는 ITQ-MC/IP 연산기 내부에서 더해져 DF로 전달된다. 또한 DF에서 필요한 파라미터는 ITQ를 통해 DF로 전달되어 DF와 VLD 또는 내/외부 메모리와의 불필요한 동작을 제거하였다.

DF는 복원된 픽셀 데이터에 대해 경계면의 데이터 성격에 따라 필터링을 통해 부드러운 이미지를 만들어 주는 역할을 한다. 복호기내의 모든 연산기가 4x4 하위블록 단위로 연산을 하는데 비해 DF는 내부 제어를 간단하게 하고 참조 메모리 접근 효율을 높이기 위해 매크로블록 단위의 연산을 한다. 따라서 데이터를 모아 놓을 공간으로 내부 SRAM을 이용한다. 또한 픽셀의 재배치를 통해 동일한 1-D 필터를 이용하여 수직 방향의 필터 연산과 수평 방향의 필터 연산을 모두 지원하고 4 개의 메모리 블록 구조를 이용하여 현재 매크로블록의 픽셀과 인접한 다른 매크로블록의 픽셀의 접근 및 저장을 효율적으로 할 뿐만 아니라 필터 연산중에 움직임

보상기의 출력 픽셀을 저장하여 더블클릭 필터와 움직임 보상기 사이의 병목현상을 제거하였다.^[10]

DMACA는 DF와 MC 그리고 data_out 장치를 위해 참조 메모리인 SDRAM에 대한 읽기와 쓰기 동작을 수행한다. 연산의 복호 동작은 파이프라인을 통해 자율적으로 일어나므로 복호에는 직접적으로 관여하지 않고 메모리에 대한 접근 제어만 하면서 필요에 따라 참조 메모리 접근에 대한 우선순위 조절을 통해 간접적으로 복호 동작에 영향을 미친다. 즉, 모든 참조 메모리 접근이 DMACA를 통해 일어나므로 어느 연산기에 우선순위를 주느냐에 따라 복호 성능이 영향을 받게 된다. 따라서 DMACA는 전체적인 복호 성능을 최대화 시키는 범위내에서 영상 출력의 흐름 제어를 담당하도록 설계하였다.

위에서 언급한 구조에 따라 H.264 복호기를 Verilog-HDL을 이용하여 설계하였다. 설계된 복호기는 시뮬레이션과 FPGA를 이용하여 동작을 검증하였다. 표 1에 복호기의 합성 결과와 기준에 발표된 복호기와의 비교 결과가 나타나 있다. 제안한 구조의 복호기는 적응적 파이프라인을 검증하기 위해 각 IP가 최적화되지 않은 상태에서 통합된 것으로 최적화되면

표 1. 설계한 H.264 복호기의 합성 및 비교 결과
 Table 1. Synthesis and comparison results of H.264 decoders.

	Proposed	[5]	[11]	[12]
Gate count	209,479	242,705	217,428	456,598
Memory [bits]	70,848	23,200	81,756	164,864
Operating frequency	25MHz (0.18um)	54MHz (0.25um)	120MHz (0.18um)	100MHz (0.18um)
Image size	480p30	CIF30	1024p30	1080HD

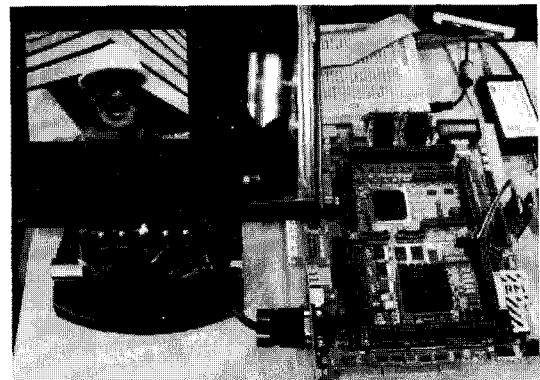


그림 7. FPGA를 이용한 H.264 복호기 검증 결과
 Fig. 7. Verification results of the H.264 decoder based on the proposed architecture using an FPGA prototyping board.

16,000 게이트 정도를 줄일 수 있다. 한편 480p 영상을 복호하기 위한 동작 주파수는 25MHz보다 더 작지만 검증 환경 때문에 25MHz로 합성하였다. 복호기별로 목표로 하는 복호 성능이 달라 직접적인 비교는 어렵지만 설계한 복호기의 면적이나 성능은 충분한 경쟁력을 가지고 있음을 알 수 있다. 또한 제어기 구조가 간단하고 각 연산기 별로 AHB 또는 핸드셰이킹 방식의 인터페이스 규격만 맞추면 연산기의 교체가 가능하여 복호기의 확장이나 성능 개선이 쉬운 것도 큰 장점이다. 그림 7에는 FPGA를 이용하여 구현된 복호기의 검증 자료가 나타나 있다.

IV. 결 론

본 논문에서는 적응적 파이프라인 구조와 AHB 네트워킹을 이용한 H.264 복호기 구조를 제안하였다. 제안한 구조에서 픽셀 데이터 재생을 위한 파이프라인은 핸드셰이킹 방식으로 외부 제어기 없이 스스로 동작하며 참조 메모리 접근은 AHB 네트워킹과 우선순위 조절을 하는 DMACA를 통해 이루어진다. 제안한 적응적 파이프라인은 영상의 특징에 따라 한 블록의 처리 시간을 가변적으로 조절하여 연산기가 쉬는 시간을 최소화하고 이에 따라 발생하는 참조 메모리 접근 문제는 DMACA의 중재 기능을 통해 해결하였다. 제안한 구조에서는 동일한 하드웨어로 평균적으로 더 높은 성능을 이끌어 낼 수 있고 연산기의 인터페이스가 표준 방식이어서 연산기 교체 및 성능 개선이 쉽다는 장점이 있으며 불필요한 동작을 억제하고 동작 주파수를 낮출 수 있어 저전력 복호기 설계가 가능하다. 제안한 구조를 검증하기 위해 H.264 복호기를 설계하여 FPGA를 이용하여 검증하였다.

참 고 문 헌

[1] Joint Video Team, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, May 2003.

[2] Chuan-Yung Tsai, Tung-Chien Chen, To-Wei Chen, Liang-Gee Chen, "Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder" Proceedings of 48th

Midwest Symposium on Circuits and Systems, Vol. 2, pp. 1199-1202, Aug. 7-10, 2005.

[3] Mythri Alle et al, "High performance VLSI implementation for H.264 Inter/Intra prediction", Proceedings of IEEE International Conference on Consumer Electronics, pp.1-2, Jan. 2007.

[4] Wen-Nung Lie, Han-Ching Yeh, Lin, T.C.-I., Chien-Fa Chen, "Hardware-efficient computing architecture for motion compensation interpolation in H.264 video coding", Proceedings of IEEE International Symposium on Circuits and Systems, Vol. 3, pp. 2136-2139, May 23-26, 2005.

[5] S. M. Park, M. Lee, S. Kim, K.-S. Shin, I. Kim, H. Cho, H. Jung, and D. Lee, "VLSI Implementation of H.264 Video Decoder for Mobile Multimedia Application," ETRI Journal, Vol. 28, No. 4, Aug. 2006.

[6] 유용훈, 이찬호, "하드웨어 구조의 H.264/AVC 가변길이 복호기 설계," 전자공학회 논문지, 제45권 SD편 제11호, pp.71-76, 2008.11

[7] 이찬호, "H.264 복호기를 위한 스케일러블 인트라 예측기 구조 설계," 전자공학회 논문지, 제45권 SD편 제11호, pp.77-81, 2008.11

[8] 이찬호, "H.264 복호기를 위한 효율적인 예측 연산기 설계," 전자공학회 논문지, 제46권 SD편 제7호, pp.47-52, 2009.7

[9] 김정현, 박황희, 이찬호, 노태문, 여순일, "저전력 H.264 Forward/Inverse Transform & Quantization 설계," 제13회 반도체학술대회논문집, pp.367-368., 2006.2.23

[10] 유용훈, 이찬호, "효율적인 메모리 관리 구조를 갖는 H.264용 고성능 디블록킹 필터 설계," 전자공학회 논문지, 제45권 SD편 제1호, pp.64-70, 2008.1

[11] T.-C. Chen, C.-Jr Lian, and L.-G. Chen, "Hardware Architecture Design of an H.264/AVC Video Codec," Proceedings of ASPDAC 2006, pp. 750-757, Jan. 24-27, 2006.

[12] T.-M. Liu, T.-A. Lin, S.-Z. Wang, W.-P. Lee, K.-C. Hou, J.-Y. Yang and C.-Y. Lee, "An 865- μ W H.264/AVC Video Decoder for Mobile Applications," Proceedings of ASSCC 2005, pp. 301-304, Nov. 1-3, 2005.

저 자 소 개

이 찬 호(정회원)

대한전자공학회 논문지 제43권 SD편 제9호 참조
현재 숭실대학교 정보통신전자공학부 교수